IN THE CLAIMS:

1. (Currently amended) A method of implementing a switch instruction in an IA64 architecture based data processing device, comprising:

receiving a call to the switch instruction, the call including one or more parameters for the switch instruction;

loading a plurality of predicate registers with values associated with a plurality of branch addresses based on the one or more parameters; and

calling an instruction associated with one of the plurality of branch addresses based on the values of the plurality of predicate registers.

2. (Original) The method of claim 1, wherein the one or more parameters includes a range of branch address, the range being defined by a high value and a low value, the method further comprising:

determining if the low value is lower than a lowpredicate;

setting a first register value to 2**(lowpredicate- low value) if the low value is lower than the lowpredicate; and

setting the first register value to 2**(lowpredicate) if the low value is not zero, where lowpredicate is a predicate register number of a lowest numbered predicate register.

3. (Original) The method of claim 2, wherein the one or more parameters includes an index into the range of branch addresses, the method further comprising:

setting a second register value equal to the index if the low value is lower than the lowpredicate; and

setting the second register value equal to a difference between the index and the low value if the low value is not lower than the lowpredicate.

4. (Original) The method of claim 3, further comprising shifting the first register value to the left by the second register value.

5. (Original) The method of claim 4, further comprising moving the first register value to the plurality of predicate registers.

6. (Original) The method of claim 5, wherein moving the first register value to the plurality of predicate registers is performed using a mask identifying predicate registers to be loaded.

7. (Original) The method of claim 1, wherein the switch instruction is a Java tableswitch instruction.

8. (Original) The method of claim 1, wherein the method is implemented in a Java Virtual Machine.

9. (Original) The method of claim 1, wherein the switch instruction is a dense switch statement in C.

10. (Original) The method of claim 4, wherein the one or more parameters includes a default address, and wherein if the first register value is shifted outside of a range of the plurality of predicate registers, an instruction associated with the default address is called.

11. (Currently amended) An apparatus for executing a switch instruction in an IA64 architecture based data processing device, comprising:

means for receiving a call to the switch instruction, the call including one or more parameters for the switch instruction;

means for loading a plurality of predicate registers with values associated with a plurality of branch addresses based on the one or more parameters; and

means for calling an instruction associated with one of the plurality of branch addresses based on the values of the plurality of predicate registers.

12. (Original) The apparatus of claim 11, wherein the one or more parameters includes a range of branch address, the range being defined by a high value and a low value, the apparatus further comprising:

means for determining if the low value is 1;

means for setting a first register value to 2**(lowpredicate-low value) if the low value is lower than the lowpredicate; and

means for setting the first register value to 2**(lowpredicate) if the low value is not lower than the lowpredicate, where lowpredicate is a predicate register number of a lowest numbered predicate register.

13. (Original) The apparatus of claim 12, wherein the one or more parameters includes an index into the range of branch addresses, the apparatus further comprising:

means for setting a second register value equal to the index if the low value is lower than the lowpredicate; and

means for setting the second register value equal to a difference between the index and the low value if the low value is not lower than the lowpredicate.

14. (Original) The apparatus of claim 13, further comprising means for shifting the first register value to the left by the second register value.

15. (Original) The apparatus of claim 14, further comprising means for moving the first register value to the plurality of predicate registers.

16. (Original) The apparatus of claim 15, wherein the means for moving the first register value to the plurality of predicate registers uses a mask to identify predicate registers to be loaded.

17. (Original) The apparatus of claim 11, wherein the switch instruction is a Java tableswitch instruction.

18. (Original) The apparatus of claim 11, wherein the switch instruction is a dense switch statement in C.

19. (Original) The apparatus of claim 14, wherein the one or more parameters includes a default address, and wherein if the first register value is shifted outside of a range of the plurality of predicate registers, an instruction associated with the default address is called.

20. (Currently amended) A computer program product in a computer readable medium for executing a switch instruction in an IA64 architecture based data processing device, comprising:

first instructions for receiving a call to the switch instruction, the call including one or more parameters for the switch instruction;

second instructions for loading a plurality of predicate registers with values associated with a plurality of branch addresses based on the one or more parameters; and

third instructions for calling an instruction associated with one of the plurality of branch addresses based on the values of the plurality of predicate registers.

21. (Original) The computer program product of claim 20, wherein the one or more parameters includes a range of branch address, the range being defined by a high value and a low value, the computer program product further comprising:

fourth instructions for determining if the low value is 1;

fifth instructions for setting a first register value to 2**(lowpredicate-low value) if the low value is lower than the lowpredicate; and

sixth instructions for setting the first register value to 2**(lowpredicate) if the low value is not lower than the lowpredicate, where lowpredicate is a predicate register number of a lowest numbered predicate register.

22. (Original) The computer program product of claim 21, wherein the one or more parameters includes an index into the range of branch addresses, the computer program product further comprising:

seventh instructions for setting a second register value equal to the index if the low value is lower than the lowpredicate; and

Eighth instructions for setting the second register value equal to a difference between the index and the low value if the low value is not lower than the lowpredicate.

23. (Original) The computer program product of claim 22, further comprising ninth instructions for shifting the first register value to the left by the second register value.

24. (Original) The computer program product of claim 23, further comprising tenth instructions for moving the first register value to the plurality of predicate registers.

25. (Original) The computer program product of claim 24, wherein the tenth instructions for moving the first register value to the plurality of predicate registers are executed using a mask identifying predicate registers to be loaded.

26. (Original) The computer program product of claim 20, wherein the switch instruction is a Java tableswitch instruction.

27. (Original) The computer program product of claim 20, wherein the computer program product is executed in a Java Virtual Machine.

28. (Original) The computer program product of claim 20, wherein the switch instruction is a dense switch statement in C.

29. (Original) The computer program product of claim 23, wherein the one or more parameters includes a default address, and wherein if the first register value is shifted outside of a range of the plurality of predicate registers, an instruction associated with the default address is called.